(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 April 2002 (11.04.2002)

PCT

(10) International Publication Number
WO 02/29719 A2

(51) International Patent Classification[7]: G06T 5/00

(21) International Application Number: PCT/DK01/00652

(22) International Filing Date: 5 October 2001 (05.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PA 2000 01493 6 October 2000 (06.10.2000) DK

(71) Applicant (*for all designated States except US*): **GENERIC VISION APS** [DK/DK]; Glentevej 67, DK-2400 Copenhagen NV (DK).

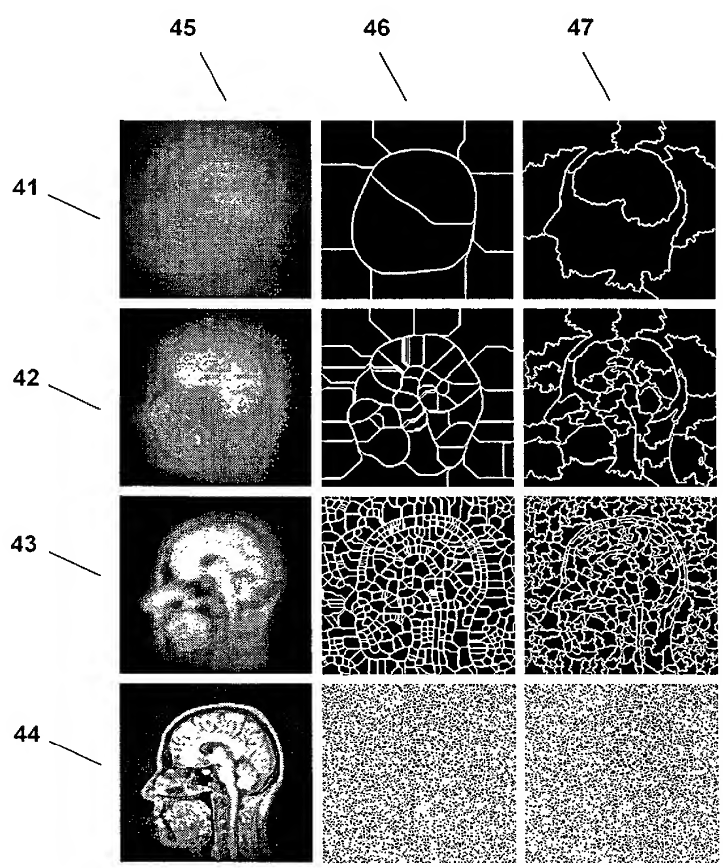
(72) Inventors; and
(75) Inventors/Applicants (*for US only*): **OLSEN, Ole, Fogh** [DK/DK]; Vintappervej 6, 1., DK-2800 Kgs. Lyngby (DK). **DAM, Erik, Bjørnager** [DK/DK]; Kronborggade 26, 5.th., DK-2200 Copenhagen N (DK). **LILLHOLM, Martin** [DK/DK]; Marstalsgade 27, 3.th., DK-2100 Copenhagen Ø (DK). **THOMSEN, Andreas, Abildhauge** [DK/DK]; Grønjordsvej 1, vær. 4418, DK-2300 Copenhagen S (DK). **JOHANSEN, Peter** [DK/DK]; Ørnebakken 72, DK-2840 Holte (DK). **NIELSEN, Mads** [DK/DK]; Jan Timanns Plads 5, DK-2791 Dragør (DK).

(74) Agent: **HOFMAN-BANG ZACCO A/S**; Hans Bekkevolds Allé 7, DK-2900 Hellerup (DK).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EC, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR MULTI-DIMENSIONAL SEGMENTATION



(57) Abstract: For use in a system for interactive segmentation of 3D images a method for the segmentation of images into semantical parts, an application of the method being medical scanning data, like for instance MR and CT, but the system being equally usable for other types of images, like for instance bifocal 3D microscopy images and 3D geological images of the underground; the method pre-computes a number of 3D segment building blocks; a user may interactively select blocks to build a segment, this may be done in virtual 2D slices of the volume or in 3D visualizations of the data, the method providng fast processing in 3D, which is advantageous for substantially instant and thereby natural interaction.

WO 02/29719 A2

LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Method and system for multi-dimensional segmentation

This invention relates to the processing of multidimensional representations of objects, and more specifically to a method for segmenting a multidimensional representation into semantic parts.

In a large number of fields, the analysis of images or other representations of objects plays an important role. In the field of medical scanning, for example, two- or three-dimensional images are generated based upon the output of a variety of scanning techniques, such as Magnetic Resonance (MR) imaging, Computed Tomography (CT) imaging, Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), ultrasound or the like. Since all of these scanning techniques have their merits and disadvantages, it may be an advantage to combine their outputs in a single multi-mode image.

Other fields where image analysis is an important aspect include, bifocal 3D microscopy images, satellite imagery and geological analyses, e.g. for the exploitation of natural resources, such as oil.

In many image analysis applications the segmentation of an image is an important pre-processing step in order to visualise and process digital images acquired for instance through medical scanning. The problem of segmentation is far from solved and existing implementations of segmentation methods are typically either manual or highly specialized automated solutions.

Manual segmentation of objects in 3D medical images made up of a number of 2D slices is based on manual identification of the objects in every slice. The

identification is typically done by hand-drawing along the object contour. In principle, manual segmentation is applicable for all segmentation tasks, but the high dependence on user interaction makes it both slow and only as accurate and reproducible as time and the user allow.

Automated special-purpose segmentation systems offer automatic identification of desired objects and thus a minimum of user interaction, but they are tailor-made for special segmentation tasks based on a priori knowledge of the object's shape and size. Automatic segmentation tools are seldom versatile and one often needs a specialized tool for each segmentation task.

US patent no. 6,064,391 discloses a method for extracting three-dimensional segments and for displaying them. The method may be used for visual inspection, and an operator may interact with the system during the segmentation process. The segmentation is done by region growing.

Multiscale segmentation has successfully been applied to a number of applications in order to select and visualize features at different scales. In multiscale segmentation an image is blurred, for example by applying a Gaussian filter, where the width of the Gaussian corresponds to a scale, i.e. to the size of details, which are removed during the blurring. Performing a segmentation corresponding to different scales, i.e. at different resolutions, facilitates the analysis of an image. While small details may be distinguished at small scales, larger structures including many details may be identified more easily at larger scales.

Multi-scale watershed segmentation divides an image into watershed basins of a dissimilarity measure such as the gradient magnitude of the original image. This method

views the intensity values of an image as a landscape with hills and valleys, the boundaries of the watersheds thus corresponding to ridges in this landscape. Hence, the boundaries of gradient watershed basins correspond to edges in the original image, i.e. areas of large gradient.

In "Multi-scale gradient magnitude watershed segmentation" by Ole Fogh Olsen and Mads Nielsen, in "Image analysis and processing," 9th International Conference ICIAP'97, Florence, Italy, edited by Alberto Del Bimbo, a method for segmenting multidimensional images is described. The method is based on watershed segmentations of a gradient representation of the original image at different scales and on a subsequent linking of segmentations at different scales.

The prior art involves the disadvantage that large amounts of data have to be stored, limiting the size and resolution of the images that can be processed on a given computer.

Another disadvantage of the prior art systems is that user interactions with visualisations of the segmentation on a computer screen are slow.

Thus an object of this invention is to provide a versatile segmentation method which may be applied to a variety of tasks.

It is a further object of the invention to provide a method with a high degree of automation.

It is another object of the invention to provide a method that allows fast visualisation and fast interaction with the visualisation of the segmentation.

It is yet another object of the invention to provide a method that allows the processing of large images with low storage capacity.

It is a further object of the invention to provide a method that is easy to use even by a user with little prior knowledge in the field of image segmentation.

It is a further object of the invention to provide a method that is applicable in two, three and more dimensions.

According to a first aspect of the invention, these and other objects are achieved when a method of segmenting a multidimensional representation of an object, the multidimensional representation comprising a plurality of indexed elements, the method comprising the steps of

processing the multidimensional representation to obtain at least a coarse and a fine segmentation corresponding to respective scales indicative of a size of details preserved by the segmentation, the coarse segmentation corresponding to a largest scale and including at least one segment, the fine segmentation corresponding to a smallest scale and including at least two segments and at least as many segments as the coarse segmentation; and

generating a representation of at least one hierarchic tree structure including a plurality of nodes and a plurality of links, wherein

- each node represents a segment of a segmentation and thereby relating to the corresponding scale of said segmentation;

- each link relates a first node related to a first scale to a second node related to a second scale;

- a root node is a node representing a segment of the coarse segmentation;

- a leaf node is a node representing a segment of the fine segmentation;

- each node corresponds to a root node of a subtree of the hierarchic tree structure where a subset of the segments of the fine segmentation corresponds to the leaf nodes of said subtree;

is characterised in that the step of generating a representation of a hierarchic tree structure includes the step of assigning a respective index of a set of ordered indices to each of the leaf nodes of the hierarchic tree structure, such that the indices of the leaf nodes of any subtree of the hierarchic tree structure constitute an interval of consecutive indices.

It is an advantage of the invention that segments which are linked to the same segment at a larger scale may readily be identified by determining a single interval instead of the individual segments, therefore providing fast access to all segments of selected portion of the image.

In a preferred embodiment of the invention the method further comprises the step of storing in a first storage means a mapping from each of the elements to a respective segment of the fine segmentation. This provides the advantage that a segment may be selected with low computational effort by selecting any of the elements of the segment.

In a preferred embodiment of the invention the step of generating a representation of a hierarchic tree structure comprises the step of recursively processing

all segments of all segmentations starting with a segment of the coarse segmentation and until all segments are processed, where the recursive processing comprises the steps of

a) processing a first segment of a first segmentation corresponding to a first scale and attempting linking the first segment to a corresponding second segment of a second segmentation with a second scale, where the second scale is a largest of said scales which is smaller than the first scale;

b) if attempting linking the first segment succeeds continuing the processing of all segments with the second segment; otherwise

c) if there is a third segment of a third segmentation with a third scale larger than the first scale, where the third segment is already linked to the first segment, continuing the processing of all segments with the third segment; otherwise

d) if there is a fourth segment of the coarse segmentation which is not linked to any segment, continuing the processing of all segments with the fourth segment.

Furthermore, the ordering of the tree structure may be achieved by recursively processing all segments of all segmentations by a method selected from the class of methods comprising pre-order, post-order and in-order traversals.

It is an advantage of this embodiment of the invention, that an ordering of the indices of the leaf nodes according to the invention is ensured during the generation of the tree structure.

In another preferred embodiment of the invention the representation of a hierarchic tree structure comprises a table where each entry corresponds to a leaf node, such that a first entry, corresponding to a first leaf node, identifies

a) a second leaf node of a smallest subtree including the first leaf node and the second leaf node, where the second leaf node has the largest index of all leaf nodes of said smallest subtree, and

b) the scale related to the root node of said smallest subtree.

Alternatively, the representation of a hierarchic tree structure may comprise a table where each entry corresponds to a node, a first entry corresponding to a third node, which is a root node of a first subtree, the first entry identifies

a) a first leaf node of the first subtree with a largest index, and

b) a second leaf node of the first subtree with a smallest index.

It is an advantage of these embodiments that the hierarchic tree structure of segmentations may be stored in a memory efficient way. It is a further advantage that the identification of related segments at other scales only requires little processing time, therefore any interaction with the segmentation, such as changing the scale of the viewed segmentation or selecting segments, is fast and therefore user-friendly.

In yet another preferred embodiment the method further comprises a step of selecting at least one portion of the

multidimensional representation, the step of selecting further comprising the steps of

a) selecting a first element;

b) selecting a scale;

c) identifying a first segment of the fine segmentation, where the first element is mapped to said first segment according to the stored mapping;

d) calculating a set of leaf nodes of a subtree having a root node related to the selected scale and having a first leaf node which represents the first segment; and

e) identifying at least one element of the multidimensional representation which is mapped according to the stored mapping to a segment represented by any of the calculated leaf nodes.

Consequently, it is an advantage of the invention that the segmentations and their hierarchic linking are stored as efficient data structures allowing a fast and memory-efficient processing. Therefore, a user interacting with the system only experiences a negligible delay between the issuing of a command, for example via a mouse click or via dragging the mouse, and the resulting change of the viewed segmentation.

In a further preferred embodiment the step of processing the multidimensional representation further comprises the steps of

a) calculating a first representation of the multidimensional representation, the first representation representing a level of variation in the multidimensional representation;

b) segmenting the first representation to obtain a watershed segmentation of the first representation, a segment of the watershed segmentation including all elements of the first representation, from which there is a path of steepest descent to a corresponding local minimum of the first representation.

This provides the advantage that the segmentation not only relies on local information at the individual element, for example a pixel, but takes global information about properties of the representation into account.

In a further preferred embodiment of the invention the method further comprises a step of selecting at least one portion of the multidimensional representation, the step of selecting further comprising the steps of

a) selecting a seed segment of the watershed segmentation;

b) assigning to at least a first and a second segment of the watershed segmentation a respective first and second rank, a selected one of the first and second ranks corresponding to predetermined properties of a set of segments on a path from the seed segment to the corresponding first or second segment;

c) sorting the first and second segment according to their respective first and second ranks;

d) selecting a number of sorted segments to be included in the selected portion of the multidimensional representation;

This provides the advantage, that the user may easily select a plurality of similar segments relating to the

same structure. This is particularly advantageous for extended, and especially elongated structures, such as blood vessels, bones, long objects, and extended mineralogical structures.

In yet another preferred embodiment of the invention the method further comprises the step of visualising at least one selected portion of the multidimensional representation, the step of visualisation further comprising the steps of smoothing the respective surfaces of a first and second selected portion by a linear filter and combining the respective smoothed surfaces to a combined smoothed surface.

It is an advantage of the invention that the successive selection of a plurality of segments or the de-selection of segments from a set of previously selected segments, may be performed with little computational effort.

In another preferred embodiment of the present invention the method further comprises the step of blurring the multidimensional representation by anisotropic diffusion to obtain at least one blurred representation corresponding to a respective scale. This gives the advantage that the blurring may be adapted to specific image analysis tasks.

In a second aspect of the invention the above objects are achieved when a method for processing a multidimensional representation of an object, the multidimensional representation comprising a plurality of indexed elements, the method comprising the steps of

processing the multidimensional representation to obtain at least a coarse and a fine segmentation corresponding to respective scales indicative of a typical size of details preserved by the segmentation, the coarse

segmentation corresponding to a largest scale and including at least one segment, the fine segmentation corresponding to a smallest scale and including at least two segments and at least as many segments as the coarse segmentation;

selecting a portion of the multidimensional representation including a first segment of a first segmentation and a second segment of a second segmentation;

calculating a contour surface of the selected portion;

is characterised in that the step of processing the multidimensional representation further includes the steps of

a) smoothing the first and second segment to obtain respective first and second smoothed segments; and

b) storing in a first storage means the first and second smoothed segments;

the step of calculating a contour surface of the selected portion further includes the steps of

a) determining an intersection of the first and second segment;

b) smoothing the intersection to obtain a smoothed intersection; and

c) combining the stored first and second smoothed segments with the smoothed intersection.

It is an advantage of the invention that it provides fast interaction with the segmented representation, for example an image. When several segments are successively

selected and their combined smoothed surface is visualised, essentially only the intersection of the latest segment with the previously selected segments has to be processes. Therefore an efficient method for interactively selecting segments is achieved.

The invention will be explained more fully below in connection with preferred embodiments and with reference to the drawings, in which:

fig. 1 shows a flow diagram of a first embodiment of the invention;

fig. 2 shows a flow diagram of the substeps of step 2 of fig. 1;

fig. 3 shows substeps of step 22 of figure 2;

fig. 4 shows a sequence of blurred versions at different scales of a 2D slice of an MR scanning and corresponding segmentations;

fig. 5 schematically shows a representation of a horizontal section of a segmented 3D image;

fig. 6 schematically shows a graphical representation of a linking tree and its representation according to a second embodiment of the invention;

fig. 7 schematically shows a graphical representation of a linking tree and its representation according to a third embodiment of the invention;

fig. 8 schematically shows the information flow during the selection of a segment according to a fourth embodiment of the invention;

fig. 9 schematically shows the selection of segments by interactive flooding according to a fifth embodiment of the invention; and

fig. 10 shows a flow diagram of a method for visualising selected portions of the segmentation according to a sixth embodiment of the invention.

Referring to fig. 1, a first embodiment according to the invention comprises the following steps:

In a first step 1 an image or another multidimensional representation of an object is imported. The multidimensional representation may be a two-, three- or higher dimensional representation, the higher dimensions representing other degrees of freedom, such as time if a time-dependant phenomenon is analysed, for example a beating heart. The representation may be a grey-scale or colour image, representing measurements in the visual or other frequency bands of the electromagnetic spectrum, or they may represent measurements of other physical properties, such as density. The images may be multi-modal images combining different measurements of the same object. Possible sources of images include medical scanning, where the image may represent a part of a human or animal body, other sources are geological or mineralogical measurements, satellite imagery, and microscopic images, where the images may represent biological tissue or other microstructures. The images may also represent photographs or drawings, for example of articles which may be selected by a user, for example by clicking on a position of a picture displayed on a computer screen. The selection by the user may initiate a subsequent transaction, such as purchasing the selected items for example via the Internet.

The representation of the object is made up of indexed elements. In a 2D image these are preferably pixels indexed by (x,y)-coordinates, and in a 3D image they are voxels indexed by (x,y,z)-coordinates. However, any other suitable representation is possible as well. Each element may represent a grey scale value, a set of colour values, or any other suitable set of attributes corresponding to the representation.

The following examples refer to three-dimensional grey-scale images, but the invention is not limited to this embodiment, and it is understood that a person skilled in the art will be able to carry out the invention in other applications.

The importing of the representation may comprise loading of an image into the memory of a computer via a standard file format, or by retrieving an image from a connected image acquisition system.

In a next step 2 building blocks for this image are created as a preprocessing step for the segmentation. The preprocessing step constructs building blocks that are aligned to the boundaries of the specific image. These building blocks are constructed with varying level of detail, and approximately 30 levels may be used. Small blocks are generated to allow segmentation of small structures in the image, larger building blocks are generated in order to allow segmentation of larger structures. The algorithm for the construction of the building blocks ensures that larger building blocks consist of a number of smaller building blocks. The location of specific boundaries between the building blocks may be maintained, but boundaries may be removed in order to merge blocks into larger building blocks. The size of the building blocks are determined by the

selected scale. The user can change the scale at any point during the interaction. A high scale means large building blocks. Preferably, between 25 and 45 scale levels are constructed depending on the image size, 45 may for example be used if the minimal coordinate dimension is more than 256 voxels. The bottom level may for example be at scale 0.5. For each scale level, the scale may be increased in predetermined steps, for example with approximately 15.

In a next step 3, a project is created that comprises the image, the building blocks, and a number of segmentations. The project data are stored on a storage media, such as a hard disk or a CD for later retrieval and further processing.

In a further processing step 4, for each segmentation, interaction with the visualized building blocks allows selection of the desired portions of the original image. The interaction may be carried out via commands, which preferably are input by a user via pointing, dragging or clicking with a computer mouse. The resulting segmentations can be inspected visually, via specific statistics such as volume and surface area, or exported for further study in other programs.

The step 2 of generating segmentations of the image comprises a number of substeps, as illustrated in figure 2. For each scale level the steps 21-23 are performed:

In step 21 a gradient scale-space is generated. Edges in the image are determined by the spatial gradient at the respective scale. In order to obtain these gradients the image is blurred, for example with a Gaussian convolution filter, which comprises a Gaussian convolution of the image $I(x)$

$$L(x,t) = \int g(x - x', t) I(x') dx' = g(x,t) * I(x)$$

where x holds the spatial coordinates, and t is the standard deviation of the Gaussian or the scale of L. Successive blurring is effectively still a blurring with a Gaussian, but with a larger spread, since

$$g(x,t_1) * g(x,t_2) = g(x, \sqrt{t_1^2 + t_2^2}) .$$

The above convolution may for example be implemented by a fourth order recursive filter implementation as described in "Recursively Implementing the Gaussian and its Derivatives", technical report 1993, INRIA - Sophia Antipolis. At a given voxel, the fourth order recursive filter needs information from the previous four voxels in all directions. This means that the values of voxels outside the image need a definition. In the known algorithms these voxels are initialised to the value zero. In a preferred embodiment according to the invention the value of the border voxels are padded into a margin, preferably 8 voxels wide, surrounding the original image. The recursive filter operation is then performed on this larger volume. The result is a volume containing the gradient squared at each image position at the desired scale.

An alternative to Gaussian blurring is non-linear diffusion. Various diffusion schemes may be used and, thus, the segmentation method may be adapted to certain classes of images, for example images including elongated structures such as blood vessels. No matter whether linear Gaussian blurring or non-linear diffusion is used, the result is a stack of volumes containing the gradient squared at each image position at the desired scales. The remaining part of the segmentation method is substantially independent of the specific blurring of the original.

The scale space image L(x,t) satisfies the heat equation

$$\partial_t L(x,t) = \frac{1}{2}\partial_{xx} L(x,t)$$

$\boxed{X}$

since the Gaussian is the Green's function of the heat equation. This gives an interpretation of an image as a temperature distribution, and the blurring may be interpreted as the diffusion of temperature due to time. In an inhomogeneous material the heat diffusion is not equal in all directions, but may follow the inhomogeneous and anisotropic heat equation:

$$\partial_t L(x,t) = \nabla\left[c(x,t)\nabla L(x,t)\right]$$

where c is the matrix-valued heat conduction. Making c a function of the image L itself, opens for non-linear diffusions schemes. If one has prior knowledge of the shape and image expressions of the objects to be recovered, a heat conduction term c may be tailored to this knowledge. This may yield edge enhancing flows ($c = f(|\nabla L|)$) where f is a monotonically decreasing positive function, or structure enhancing flows. For example the Perona-Malik flow (see "Scale-space edge detection using anisotropic diffusion" by P. Perona and J. Malik, IEEE Transactions on PAMI, 12(7):629-639) gives a substantial reduction of the number of necessary interactions for segmenting white and grey matter in 2D MR brain images. The use of structure enhancing flows (see "Anisotropic diffusion in image processing" by Joachim Weickert, Teubert, Stuttgart) are decreasing the number of necessary interactions when segmenting very elongated structures like vessels. Level set methods (see "Axions and fundamental equations of image processing", by L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, Arch. for rational Mathematics, 123(3):199-257, September, 1993) may be used for creating invariant flows, and may

be better for situations where images are of affine or projective mapped images.

Alternatively, other dissimilarity measures than the gradient magnitude squared may be used, for example higher order derivatives, other monotonous functions of the length of the gradient, or statistical measures such as moments of an intensity distribution.

Step 22 in fig. 2 comprises the construction of watershed catchment basins based on the volumes containing the gradient squared at each image position at the desired scales. A catchment basin is called a building block or a segment.

Step 24 comprises the linking of catchment basins to a hierarchic tree structure. The level with the smallest scale is denoted the localization level. The remaining levels are all linked to the level below. Thereby the large building blocks get the precision in the localization of the borders of the building blocks that the localization level provides. The linking connects the hierarchic tree structure or linking tree with the index volume. The index volume is a volume with labelled catchment basins from the localization level. It contains information about the basic building block to which a voxel belongs, as will be described in connection with fig. 5. The linking tree determines how the basins are merged together into larger regions at the higher levels.

The linking between two consecutive levels, a low level and a high level, may be performed as follows:

For each voxel, it is determined which basins it belongs to at both the low level and the high level. During the traversal of the volumes, a list of the overlapping basins at the high level is constructed for each basin at

the lowest level. The list contains the number of overlapping voxels.

For each basin at the low level, it is determined which basin at the high level has the highest number of overlapping voxels. The child basin at the low level is linked to this parent basin at the high level in the linking tree.

The linking tree may be constructed as a list of lists. For each level a list of the links to the next levels is kept. These lists are sorted such that the tree does not contain crossing branches.

Since the linking is done from basins at low scale to basins at higher scale, it is ensured that a basin at high scale is always the parent of a basin at the lower scale.

The index volume and the linking tree are represented in the following data structures: the index volume, the building blocks, and the linking tree. These are described below in connection with fig. 5-7.

In addition to the scale levels from the localization level up to the highest scale level an extra level may be added. The building blocks at this level are the individual voxels. This enables the user to select/de-select structures of the desired detail.

Now referring to fig. 3, the partitioning of the volume into catchment basins comprises a number of substeps:

In step 31 each image voxel is annotated with the direction of steepest descent. This is defined as the neighbouring voxel with the largest intensity decrease (with due consideration of possible different voxel sizes in the different coordinate directions). A non-boundary

voxel in three dimensions has 26 neighbours. It is understood that the term intensity covers intensity values of grey-scale images as well as its generalisation in colour images and multi-modal images. Voxels in local minima are assigned with no descent direction. Exceptions are voxels with one or more neighbours with the same value. If one or more of these neighbours are located in a direction of increasing coordinate values, an arbitrary among these voxels may be used as the steepest descent direction. This special case handles plateaus with equal intensity values, which are primarily present in artificial images.

In step 32, each image voxel is annotated with a dummy label meaning that it has not yet been assigned to a catchment basin.

For each voxel, a search for the corresponding catchment basins is then initiated. From the particular voxel the path of steepest descent is followed (step 36) and traced (step 34) in a stack until a voxel is reached that is annotated as a local minimum (step 35), or that already has been annotated with a basin label (step 37). If a local minimum is reached, the voxels in the path stack are annotated with a new basin label (step 38). If a voxel with a basin label is reached, this label is used for the voxels in the path stack.

Alternatively, other known search algorithms for the catchment basins may be used. The steepest descent path has the advantage that the length of the path is minimized and thereby also the memory requirements for the path stack. The result of the process is a volume where each voxel is annotated with the corresponding catchment basin label. A 3D watershed is the surface which separates the catchment basins.

The result of this processing is a set of disjoint building blocks, a linking tree and an index volume.

In fig. 4, the result of the embodiment illustrated in fig. 2 for a 2D slice of a 3D image of an MR scan of a head is shown. The original image is 256x256 pixels. The leftmost column 45 shows the slices blurred with a Gaussian filter with scales from bottom to top of σ=0.75, 4.59, 11.9, 30.9 pixels, respectively. The middle column 46 is the watershed segmentation of the left column. In the he right column 47 the segment boundaries correspond to the boundaries of the respective segments at the finest scale or localisation scale, as tracked by the linking.

Now referring to fig. 5, the index volume according to a preferred embodiment of the invention is a volume of the same dimensions as the original input volume. Each voxel of the index volume is the index of the building block that the voxel is part of. Fig. 5 illustrates a slice 51 of the index volume. The axes of the index volume are called x, y and z, respectively. The index volume is represented as a sequence of lines. Each line 52 consists of all voxels with fixed x and y coordinate and with varying z coordinate. Each line 52 is finally represented by its run length code. Each voxel contains the identifier of a segment at the localisation scale. In fig. 5 it is assumed that the slice 51 comprises voxels corresponding to building blocks 4, 7, and 11, respectively. This is indicated by the areas 51a-c in fig. 5. Each line 52 with constant x and y contains a sequence of integers 52a-c. The lines are preferably encoded with a variant of the known run length encoding.

Now referring to fig. 6, the linking tree 601 according to a second embodiment of the invention is the merging of

building blocks at higher scales. In the linking tree each node is a building block. Each leaf is a basic building block. The tree 601 can be displayed on a plane without crossing branches. If there are more than one segments at the largest scale, the tree structure comprises several disconnected trees, a so-called forest. By sorting the basic building blocks, preferably by a depth-first scheme such as in a pre-order run of the linking tree, the leafs covered by each node will be an interval in the pre-order sorted list. Here, the term pre-order run refers to a known technique for recursively processing the nodes of a tree in which the root is processed first, then any subtrees. It is understood that other processing schemes may also be used, for example post-order or in-order traversals. By exploring this the building blocks and the linking information may be represented as just one list of building blocks and links to the next scale in each block. The linking mechanism described above has defined a set of segments for each one of a number of scales 602. The scales 602 may be identified by a number, the smallest scale as scale 1, also called the localisation scale. The scale immediately above is number 2, etc, until the largest scale which has number S_{max}. The linking mechanism associates with each segment at scale S (where $1< S \leq S_{max}$) at least two segments at scale less than S_{max}. A segment at scale 1 is called a basic segment. Let the number of basic segments be M. Each basic segment may be identified by a number between 1 and M. Each segment at a scale larger than 1 is represented by an inner node in the forest described below. The links between segments at different scales constitute a forest. The trees of the forest are ordered, and each tree may be traversed in pre-order or, alternatively, for example in post-order.

When a user interacts with the building blocks two parts of information are available: the selected basic segment and the selected scale. The purpose of the linking table is to provide the unique interval of basic segments that include the given basic segment at the given scale.

According to the embodiment of figure 6, the tree is represented by tables. The table entries each contain two pairs of numbers: the right entries 603a-h, 604a-h and the left entries (not shown). The right and left entries allow the upper, respectively lower, limit of the interval to be established. The entries of the tables are indexed by the M leaves of the forest. The leaves of a node of scale S in a tree correspond to an interval of leaf identifiers. This means that the corresponding segment at scale S consists of all basic segments in the leaf interval. Let I be the intervals of identifiers of basic segments which constitute a segment at a higher level. Two intervals from I are either nested or disjoint. Thus, in order to find the segment at scale S which contains basic segment L, it is sufficient to find the node at scale S whose interval contains L.

The right entries 603a-h, 604a-h are used to establish the upper limit for the desired interval. Given a leaf node L, the minimal upper limit for an interval is the node itself. The right pointers 603a-h point to the next possible upper limit for an interval. This is the next leaf to the right which is the rightmost leaf in a subtree. The corresponding right scales 604a-h tell how high a scale is needed to extend the interval to have the right pointer as the upper limit. The table entries for the right pointers 603a-h and the right scales 604a-h may be constructed from the linking tree as follows: The right table is constructed starting by its last entry.

The general construction of the table representation can be described as follows:

The right table entries are filled from the highest index towards the lowest index one by one. Initially the right pointer 603h and the right scale 604h at the highest index are set to nil, and all nodes along the parent path from the leaf 605 with highest index to the root 607 are set to that index. Starting by the highest index minus one, at leaf L=N, its parent chain is followed until the first previously marked node. Let it be at scale S and let its mark be M. Since initially the root 607 was marked, it is certain that a previously marked node will be reached when following the parent chain from a leaf. Along the way unmarked nodes receive the mark N. The right pointer at index N is set to M and the right scale at index N is set to S.

With reference to the example of fig. 6, the links in the tree are followed starting at leaf 605 with L=8. Its parent 606 is marked by 8, its parent again by 8, until the root 607 has been marked by 8. The right pointer 603h and the right scale 604h are set to the nil pointer. Next entries 603g and 604g with index 7 have to be filled. The parent 606 of leaf 608 with L=7 has already been marked. The right pointer 603g of entry 7 is set to the parent's mark which is 8, and the right scale 604g is set to the parent's scale, which is 2. Next entry 6 is filled. The parent 610 of leaf 609 with L=6 has not been marked. It is marked by 6, and its parent link 611 is followed. The next parent 612 (which is at scale 3) has not been marked previously, and is marked by 6. Its parent again is the root 607 of the tree which is at scale 5 and has been marked by 8. Since it is the first marked node in the parent chain of leaf 609 with L=6, its mark (which is 8) is filled into the right pointer entry 603f at index 6.

The right scale entry 604f at index 6 is set to the scale of the first previously marked node (which in the example is scale 5). The right scale is set to 5. Next, the parent chain of leaf 613 with L=5 is followed until a marked node is met. The parent 610 of leaf 613 (which is located at scale 2) has the mark 6. Thus, the right pointer 603e at index 5 is set to 6 and the right scale 604e at index 5 is set to 2. Leaf 614 with L=4 has a previously unmarked parent 615, which is now marked 4. Its parent 612 again is located at scale 3 and has the mark 6. Thus the right pointer 603d at index 4 is set to 6, and the right scale 604d at index 4 is set to 3. The parent 615 of leaf 616 with L=3 has been marked by 4, and its scale is 2. Hence the right pointer 603c at index 3 is set to 4 and the right scale 604c at index 3 is set to 2. The parent 618 of leaf 617 with L=2 is unmarked, it is marked by 3. That node's parent 612 is at scale 3 and is marked 6. Hence the right pointer 603b at index 2 is set to 6 and the right scale 604b at index 2 is set to 3. Finally, the parent 618 of leaf 619 with L=1 is at scale 2 and has been marked. Thus, the right pointer 603a at index 1 is set to 2, and the right scale 604a at index 1 is set to 2.In the following, still referring to fig. 6, the use of the table representation to find the relevant interval for a building block at scale S which contains basic building block L is illustrated. When the operator has chosen a scale and has selected a basic building block, the right table and the left table is used to compute the interval of basic building blocks that constitute the building block at scale S which contains basic building block L. The right table is used to find the right end point of the interval and the left table is used to find the left end point of the interval.

Explained in terms of the linking tree, two steps are performed in order to compute the right end point of the

interval: First, the tree is ascended until the largest scale which is not exceeding S. Second, the tree is descended following the rightmost pointer until one reaches a leaf. The number of that leaf is the number of the rightmost building block in the selected building block at scale S.

According to the invention, the same result can be computed significantly more efficiently, only using the tables and not using the links in the tree. This follows, since the mark at each internal node is the index of its rightmost leaf, as described above.

With reference to figure 6, this will be illustrated by an example assuming that the operator has selected scale S=4 and basic building block L=3. A table look-up in table 604a-h shows that the right scale at index 3 is 2, as indicated by entry 604c. Since 2 is not larger than S, the parent link 620 of the parent 615 of leaf 616 with L=3 is followed. But this parent link 620 is identical to the parent link of the rightmost child 614 of node 615. However, this rightmost child 614 is exactly the information (L=4) contained in the right pointer entry 603c at index 3. This tells us that the parent link we want to follow is also the parent link 620 of leaf 614 with L=4. Thus one repeats the computation at leaf 614 with L=4. The right scale entry 604d at index 4 is still not larger than S=4, which means that one follows the right pointer 603d at index 4 containing the value L=6. At index 6 the right scale 604f is 5, which is greater than S=4. Thus, one has to back up a step and use the right pointer 603e with index 5 as the right end point of the interval, i.e. the leaf 609 with index 6. Hence, the above example illustrates that the right endpoint of the desired interval at a given scale S which contains a

certain building block may efficiently be computed based on the constructed tables.

Similarly, the entries of the left tables may be used to find the left end point of the interval without having to use the actual linking tree.

The table entries for the left pointer and the left scale are constructed similarly to the right entries. Given a leaf node L the predecessors are visited upwards in the tree until a node N_1 is reached which has a leftmost child which has not been visited. The scale S_1 at which this node N_1 is located is the left scale for the leaf node L. The leftmost child of the node N_1 is called L_1. This leaf L_1 is the lower limit for the interval including the leaf L at scale S_1. The leaf L_1 is the left pointer for the leaf node L.

Referring to fig. 7, according to a third embodiment of the invention, the hierarchic tree structure 701 may be implemented by level coding. An implementation of the linking tree 701 may be done by representing each of the levels 702 of the tree 701 as two sequences: The interval's right and left endpoints. This implementation constructs for each scale two tables 703a-d and 704a-d, respectively, which contain the right respectively left interval end points. Preferably, the tables are constructed by the same traversal mechanism, which is used in the implementation by right and left pointers.

In order to construct the level coding right tables, all tree nodes are marked by their right-most leaf L as follows: All nodes along the parent path from the leaf with highest index to the root are set to that index. All right tables are initialised to contain the highest index. Starting by the highest index minus one, at leaf L=N, its parent chain is followed until the last

previously unmarked node. If no such node exists, no additions to the right tables is made that contains leaf number L=N. Otherwise, let it be at scale S and let its mark be M. The leaf number of L is added to all right tables at scale less than or equal to S.

With reference to figure 7, the links in the tree are followed starting at leaf 718 with L=8. Its parent 719 is marked by 8, its parent again by 8, until the root 720 has been marked by 8. All right tables 703a-d are initialised to contain 8. Next, entry 7 is considered. The parent 719 of leaf 721 with L=7 has already been marked. Thus, leaf number 7 is not added to any of the right tables. Next entry 6 is considered. The parent 723 of leaf 722 with L=6 has not yet been marked. It is marked by 6, and its parent link 724 is followed. The next parent 725 (which is at scale 3) has not been marked previously, and is now marked by 6. Its parent again is the root 720 of the tree which has been marked. Since node 725 is the last unmarked node in the parent chain of leaf 722 with L=6, leaf number 6 is added to all right tables at scale 3 and less, i.e. to tables 703c-d. Next, the parent chain of leaf 726 with L=5 is followed until a marked node is met. The parent 723 of leaf 726 with L=5 (which is located at scale 2) has the mark 6. Thus, leaf number 5 is not added to any right table. Leaf 727 with L=4 has a previously unmarked parent 728 at scale 2, which is now marked 4. Leaf number 4 is added to the right table 703d at scale 2. The parent 728 of leaf 729 with L=3 has been marked by 4, and 3 is not added. The parent 731 of leaf 730 with L=2 is at scale 2 and is the last previously unmarked in the parent chain. It is now marked by 2 and leaf number 2 is added to the right table 703d at scale 2. Finally, the parent 731 of leaf 732 with L=1 has previously been marked, and leaf number 1 is, therefore, not added to any of the right tables.

Similarly, in order to construct the level coding left tables 704a-d , first all tree nodes are marked by their left-most leaf L as follows. All nodes along the parent path from the leaf 732 with lowest index to the root 720 are set to that index. All left tables are initialised to contain the lowest index. Starting by the lowest index plus one, at leaf L=2, its parent chain is followed until the last previously unmarked node. If no such node exists, no additions to the left tables is made that contains leaf number L=N. Otherwise, let it be at scale S and let its mark be M. The leaf number of L is added to all left tables at scale less than or equal to S.

In fig. 7, to the right of the tree the right tables 703a-d, one for each scale greater than 1, are shown. To the left of the tree, the left tables 704a-d are shown. To find the right end point of an interval that contains basic building block n, one chooses the right table at the proper scale and finds the smallest index for which the entry is greater than or equal to n. To find the left end point one chooses the left table at the same scale and finds the largest index for which the entry is smaller than or equal to n.

In an alternative implementation of the linking tree 701 one associates two tables at each scale, in which the left and right endpoints can be found by direct indexing with the basic building block number. This implementation will use a memory amount of

2 x #scales x #basic building blocks

voxels. The implementation by left and right pointers according to fig. 6 and the implementation by level coding according to fig. 7 both have a computer memory requirement which is

4 x #basic building blocks

voxels. This means that the proposed implementations both will be more efficient if only the number of scales exceeds two. One may expect that the number of building blocks will increase as n^3 where an image is n x n x n. This means that for 16 scales the preferred embodiment can handle 2n x 2n x 2n images when the alternative implementation can handle only n x n x n images.

According to a fourth embodiment of the invention, an arbitrary number of original image volumes with corresponding building blocks and segmentations may be visualized. The modules of the projects may be hidden or displayed individually. Furthermore, the modules may be translated, rotated, and zoomed in order to focus on a desired region of interest. The scale of the buildings blocks may be changed by the user at any time during the interaction. The individual building blocks may be selected and de-selected and thus facilitate the sculpting of the desired image structure.

Selection and de-selection of a building block at a given scale may be done by adding or removing an interval of basic building blocks, respectively. The interval is found from the relevant subtree of the linking tree. For visualization the segmentation surface is smoothed.

Referring to fig. 8, the user may select the scale at which he or she wants to select a building block. The user selects the building block by pointing with a pointing device at a position within the image displayed on a computer screen and the resulting segmentation may be computed by a computer program in a sequence of steps. According to the fourth embodiment of the invention, the computation may comprise the following steps:

In a first step 81 the world coordinates of a voxel are computed. One computation is performed when an object is to be added to the existing segmentation. Another computation is performed when the object is to be removed from the existing segmentation. According to this embodiment of the invention, when the user wants to add a segment using the 3D visualization on the screen, he or she may select a pixel on the screen. The user has previously preset the scale. The selected screen position defines the user's line of sight. This line of sight is intersected by the interpolated surface of the partial segmentation. The 3D intersection point which is closest to the user defines a line through the intersection point along the normal to the interpolated surface. This line intersects a set of voxels. The first voxel counted from the intersection point along the line of sight towards the user is selected. At the same time the voxel should not belong to the partial segmentation. That voxel is called the seed voxel.

When a segment should be removed from the current segmentation, the seed voxel may be chosen in a similar manner. However, now the voxel should belong to a part of the partial segmentation.

It is understood that other methods for calculating the world coordinates may be used, instead.

In a next step 82 the basic building block is found. Given a seed voxel with coordinates (x,y,z) one may retrieve the number of the basic segment which contains the seed voxel from the run-length encoded line with coordinates (x,y) of the line representation of the basic segments in the index volume 83. The basic segment number can be determined directly from the run length encoded line.

In a next step 84, the interval of basic building blocks is found based upon a representation of the linking tree 85. For example, if the linking tree is represented by left and right pointers according to the embodiment of fig. 6, the interval end points may be determined from a basic segment number n as follows: From a table entry n in the tree representation of the linking structure one follows first the right leaf links of the table until the proper scale is reached. This yields leaf L_{upper} as the table index at which the search terminates. Next one follows the left leaf links of the table representation of the segment linking until the proper scale is reached. This yields L_{lower} as the table index at which the search terminates. The basic segments are all basic segments in the interval between and including L_{lower} and L_{upper}.

Other schemes for determining the interval are possible. For example, if the linking tree is represented by level coding according to the embodiment of fig. 7, the interval end points may be determined from a basic segment number n as follows. The right table at the proper scale is selected, the smallest index for which the entry R is not smaller than n is determined. The right interval end point is R. Similarly, the left table at the proper scale is selected, and the smallest index for which the entry L is not greater than n is determined. The left interval end point is L.

In a next step 86, all voxels of the segment are determined. This computation has access to the coordinates (x,y,z) of the seed voxel as determined in step 81. The coordinates (x,y) determine a line parallel to the z-axis in the index volume 83. According to this embodiment of the invention a spiral path is traversed in the x,y-plane with (x,y) as centre. At each voxel on the spiral path the line which is perpendicular to the x,y-

plane is searched for voxels that belong to the segment interval. The exact shape of the spiral path is not essential, for example a path that proceeds in squares of increasing size may be used. The search terminates when no voxel has been included for an entire winding of the spiral.

Finally the representation may be changed as described in connection with fig. 10.

Now referring to fig. 9, an alternative way of selecting segments according to the invention is by Interactive Flooding (see "Multiscale Morphological Segmentations Based on Watershed, Flooding, and Eikonal PDE" by Fernand Meyer and Petros Maragos, in "Lecture Notes in Computer Science", Volume 1682, 1999, edited by Mads Nielsen, Peter Johansen, Ole Fogh Olsen and Joachim Weickert). Given a selected partial segment, one may wish to select nearby segments of similar image structure. In fig. 9 a slice 91 of a segmented image is shown schematically. Each segment of the slice 91 is labelled by a letter from 'A' to 'R'.

The selection of segments may be done by the following interactive flooding algorithm:

- Select a seed point and thus a corresponding seed segment. In the example of fig. 9 this is assumed to be the segment L.

- Sort all segments in the order in which they would become wet if the seed point were a water source.

- chose how many of the ordered segments should be selected. This may be accomplished by an interactive slider.

The step of sorting the segments may be performed iteratively according to the following algorithm:

A boundary between two segments is assigned the value of the minimal gradient magnitude of all points on the boundary. For some of the boundaries between segments in the example of fig. 9, the minimal gradient magnitude, corresponding to smallest height of the corresponding ridge in the intensity landscape, is indicated by numbers. Alternatively, this value may be combined with statistics about the two neighbouring segments. Also other numerical information about the two neighbouring segments may be used.

According to the embodiment of the invention illustrated in fig. 9, two lists are maintained: a segment list 93 contains the flooded segments, and a boundary list 94 contains the boundaries between a flooded and a non-flooded segment. The values on the boundary list are sorted according to their value in increasing order. The lists are established after selecting the seed point. The algorithm then iterates as follows:

1. Put the segment corresponding to the smallest value on the boundary list at the end of the segment list.

2. Remove all boundaries adjacent to that segment from the boundary list.

3. Insert the boundary segments between the segment and its non-flooded neighbours into the sorted boundary list.

Referring to the example of fig. 9, four iterations 92 of the method according to this embodiment of the invention are shown. In the first iteration the segment list 93 is initialised with the seed segment L, and the boundary list 94 comprises the sorted boundaries of the seed

segment. The next segment to be flooded would in this example be segment K, as the boundary between K and L has the smallest value. Hence, in iteration 2 segment K is appended to the segment list 93. According to the second step of the algorithm, the boundary between K and L is removed from the boundary list 94, and the boundaries between K and F, J, and P, respectively, are inserted in the boundary list.

In iteration 3, the next segment to be flooded would be segment M, and the algorithm continues accordingly. Fig. 9 shows the sequence in which the catchment basins are flooded for four iterations of the algorithm. At each iteration 92 a new segment joins to the flooded area. The flooded segments are kept in the segment List 93. The minimum image values on exterior boundary of segments on the segment list 93 are kept in the boundary list 94.

If the operator judges that a last segment should not have been included, it may be removed again from the segment list, for example by adjusting an interactive slider governing the number of segments to be included in the segment list 93. In that case the boundary list is adjusted accordingly. It is an advantage of this embodiment, that the lists only have to be established once after selection of a seed voxel. Therefore, the subsequent interactive adjustment of the number of segments included is fast.

Now referring to fig. 10, a smoothed segmentation may be represented as floating point volumes, for example using the same 2x2x2 cubes as the building blocks. The Addition of a list of smoothed building blocks may be done by performing an arithmetic addition operation "+" on the segmentation and each of the building blocks. To remove a building block arithmetic subtraction "-" is used. The

segmentations may be represented as binary volumes using the same 2x2x2 cubes as the building blocks. Addition of a list of building blocks may be done by performing a binary "OR" operation on the segmentation and each of the building blocks. To remove a building block the binary "NAND" operation is used.

It is possible to do the smoothing of the segmentation building blocks prior to the interaction phase. Thereby the interaction is potentially faster. The cost is the necessity for more data in the project.

The binary segmentation may be blurred using a small convolution kernel, where a 3x3x3 or 5x5x5 kernel may be sufficient). It is possible to perform pre-smoothing for fast selection of segments. The scenario is as follows: a surface rendering of a segmentation is presented to the user. The user selects a segment to join to or remove from the previous partial segmentation. As a consequence, the surface rendering is updated. In order to perform a surface rendering the surface position is given in a discretised form. Furthermore the surface normals should be known to assign the correct colours. A segment is a collection of cubic volumes (voxels). If the surface normal is assigned directly to these, the object will have an appearance as built of small cubes. If, however, the binary volume which represents the segmentation is smoothed before a contour surface is constructed, the segment will appear smooth. For this reason both the partial segmentation and all building blocks are preferably smoothed prior to selection of a building block.

Preferably, candidate volumes may be smoothed off-line before the interaction. Then interaction becomes faster since only a small amount of smoothing is part of the

interaction loop. The algorithm according to the embodiment of the invention illustrated in fig. 10 is as follows:

In step 101 all building blocks are smoothed prior to the user's selection of a building block, and the smoothed building blocks are stored on a storage media for later fast retrieval.

Upon selection 102 and during step 104, the smoothed building block is added to or subtracted from the smoothed volume of the partial segmentation. Here, addition is arithmetic addition, subtraction is arithmetic subtraction. The building blocks are no longer binary valued due to the blurring.

Subsequently, a new contour surface is computed in step 105. The computation of a new contour surface may be performed by finding the intersection between the binary volumes of the segment and the partial segmentation by a binary AND. This intersection is preferably smoothed by a linear filter. Preferably, the smoothing is only performed on the part of the volume where the filter support intersects the intersection. The results are identical to on-line smoothing. This can been seen as follows. Let F be the linear filter, let S be the selected segment and let P be the partial segmentation. Let "+" mean arithmetic addition, let "-" mean arithmetic subtraction, and let • mean binary AND.

Then in case of the addition of a building block, $(S+P-S\bullet P)$ is the same as the binary OR of the binary volumes representing the segment and the partial segmentation. $F(S)$ is the smoothed segment, $F(P)$ is the smoothed partial segmentation, and $F(S+P-S\bullet P)$ is the result which one would like to obtain. When the filter F is linear, one has $F(S+P-S\bullet P)=F(S)+F(P)-F(S\bullet P)$. This equation

implies that one gets the wanted result by arithmetic addition of the two smoothed volumes and arithmetic subtraction of the smoothed intersection between the two binary volumes. This is consistent with the general description above, since the intersection is the subvolume where both volumes are different from zero.

In the case of the subtraction of a building block F(S-S•P) is the result which one would like to obtain. When the filter F is linear, one has F(S-S•P)=F(S)-F(S•P). This equation implies that one gets the wanted result by arithmetic subtraction of the contour surface of the intersection from the contour surface of the partial segmentation.

According to the embodiment of the invention illustrated in fig. 10, rendering may performed in two steps. The smoothed surface is traversed to produce a triangulation of the surface. Then the normals are computed at the vertices shared by the triangles and formatted for graphic display. Triangulation is performed by creating a contour surface of the blurred segmentation. If precomputed blurring of the building block is not used, the segmentation is blurred prior to triangulation. The triangulation is preferably in cubes of 16x16x16 voxels. An algorithm similar to the known marching cubes algorithm may be used to produce the triangles which are collected in a temporary volume, for example a 19x19x19 volume. From this temporary volume the shared vertices are identified and the normal at each vertex is computed as the average of the normals of the triangles that use the vertex. The vertices, normals and triangles are then formatted in a way suitable for the graphics hardware on the platform used.

CLAIMS

1. A method of segmenting a multidimensional representation of an object, the multidimensional representation comprising a plurality of indexed elements, the method comprising the steps of

processing the multidimensional representation to obtain at least a coarse and a fine segmentation corresponding to respective scales indicative of a size of details preserved by the segmentation, the coarse segmentation corresponding to a largest scale and including at least one segment, the fine segmentation corresponding to a smallest scale and including at least two segments and at least as many segments as the coarse segmentation; and

generating a representation of at least one hierarchic tree structure including a plurality of nodes and a plurality of links, wherein

- each node represents a segment of a segmentation and thereby relating to the corresponding scale of said segmentation;
- each link relates a first node related to a first scale to a second node related to a second scale;
- a root node is a node representing a segment of the coarse segmentation;
- a leaf node is a node representing a segment of the fine segmentation;
- each node corresponds to a root node of a subtree of the hierarchic tree structure where a subset of the segments of the fine segmentation corresponds to the leaf nodes of said subtree.

c h a r a c t e r i s e d in that

the step of generating a representation of a hierarchic tree structure includes the step of assigning a respective index of a set of ordered indices to each of the leaf nodes of the hierarchic tree structure, such that the indices of the leaf nodes of any subtree of the hierarchic tree structure constitute an interval of consecutive indices.

2. The method according to claim 1, c h a r a c t e r i s e d in that the method further comprises the step of storing in a first storage means a mapping from each of the elements to a respective segment of the fine segmentation;

3. The method according to any one of the claims 1 and 2, c h a r a c t e r i s e d in that the step of generating a representation of a hierarchic tree structure comprises the step of recursively processing all segments of all segmentations starting with a segment of the coarse segmentation and until all segments are processed, where the recursive processing comprises the steps of

a) processing a first segment of a first segmentation corresponding to a first scale and attempting linking the first segment to a corresponding second segment of a second segmentation with a second scale, where the second scale is a largest of said scales which is smaller than the first scale;

b) if attempting linking the first segment succeeds continuing the processing of all segments with the second segment; otherwise

c) if there is a third segment of a third segmentation with a third scale larger than the first scale,

where the third segment is already linked to the first segment, continuing the processing of all segments with the third segment; otherwise

d) if there is a fourth segment of the coarse segmentation which is not linked to any segment, continuing the processing of all segments with the fourth segment.

4. The method according to any one of the claims 1 through 3, characterised in that the step of generating a representation of a hierarchic tree structure comprises the step of recursively processing all segments of all segmentations by a method selected from the class of methods comprising pre-order, post-order and in-order traversals.

5. The method according to any one of the claims 1 through 4, characterised in that the representation of a hierarchic tree structure comprises a table where each entry corresponds to a leaf node, such that a first entry, corresponding to a first leaf node, identifies

a) a second leaf node of a smallest subtree including the first leaf node and the second leaf node, where the second leaf node has the largest index of all leaf nodes of said smallest subtree, and

b) the scale related to the root node of said smallest subtree.

6. The method according to any one of the claims 1 through 4, characterised in that the representation of a hierarchic tree structure comprises a table where each entry corresponds to a node, a first

entry corresponding to a third node, which is a root node of a first subtree, the first entry identifies

a) a first leaf node of the first subtree with a largest index, and

b) a second leaf node of the first subtree with a smallest index.

7. The method according to any one of the claims 1 through 6, characterised in that the method further comprises a step of selecting at least one portion of the multidimensional representation, the step of selecting further comprising the steps of

a) selecting a first element;

b) selecting a scale;

c) identifying a first segment of the fine segmentation, where the first element is mapped to said first segment according to the stored mapping;

d) calculating a set of leaf nodes of a subtree having a root node related to the selected scale and having a first leaf node which represents the first segment; and

e) identifying at least one element of the multidimensional representation which is mapped according to the stored mapping to a segment represented by any of the calculated leaf nodes.

8. The method according to any one of the claims 1 through 7, characterised in that the step of processing the multidimensional representation further comprises the steps of

a) calculating a first representation of the multidimensional representation, the first representation representing a level of variation in the multidimensional representation;

b) segmenting the first representation to obtain a watershed segmentation of the first representation, a segment of the watershed segmentation including all elements of the first representation, from which there is a path of steepest descent to a corresponding local minimum of the first representation.

9. The method according to claim 8, c h a r a c t e r i s e d in that the method further comprises a step of selecting at least one portion of the multidimensional representation, the step of selecting further comprising the steps of

a) selecting a seed segment of the watershed segmentation;

b) assigning to at least a first and a second segment of the watershed segmentation a respective first and second rank, a selected one of the first and second ranks corresponding to predetermined properties of a set of segments on a path from the seed segment to the corresponding first or second segment;

c) sorting the first and second segment according to their respective first and second ranks;

d) selecting a number of sorted segments to be included in the selected portion of the multidimensional representation;

10. The method according to any one of the claims 1 through 9, c h a r a c t e r i s e d in that the method

further comprises the step of visualising at least one selected portion of the multidimensional representation, the step of visualisation further comprising the steps of smoothing the respective surfaces of a first and second selected portion by a linear filter and combining the respective smoothed surfaces to a combined smoothed surface.

11. The method according to any one of the claims 1 through 10, c h a r a c t e r i s e d in that the multidimensional representation is a three-dimensional image.

12. The method according to any one of the claims 1 through 11, c h a r a c t e r i s e d in that the method further comprises the step of blurring the multidimensional representation by anisotropic diffusion to obtain at least one blurred representation corresponding to a respective scale.

13. A method for processing a multidimensional representation of an object, the multidimensional representation comprising a plurality of indexed elements, the method comprising the steps of

processing the multidimensional representation to obtain at least a coarse and a fine segmentation corresponding to respective scales indicative of a size of details preserved by the segmentation, the coarse segmentation corresponding to a largest scale and including at least one segment, the fine segmentation corresponding to a smallest scale and including at least two segments and at least as many segments as the coarse segmentation;

selecting a portion of the multidimensional representation including a first segment of a first

segmentation and a second segment of a second segmentation;

calculating a contour surface of the selected portion;

c h a r a c t e r i s e d in that

the step of processing the multidimensional representation further includes the steps of

a) smoothing the first and second segment to obtain respective first and second smoothed segments; and

b) storing in a first storage means the first and second smoothed segments;

the step of calculating a contour surface of the selected portion further includes the steps of

a) determining an intersection of the first and second segment;

b) smoothing the intersection to obtain a smoothed intersection; and

c) combining the stored first and second smoothed segments with the smoothed intersection.

14. The method according to claim 13, c h a r a c t e r i s e d in that the step of processing the multidimensional representation further comprises the step of

generating a representation of at least one hierarchic tree structure comprising a plurality of nodes and a plurality of links, wherein

- each node represents a segment of a segmentation and thereby relating to the corresponding scale of said segmentation;

- each link relates a first node related to a first scale to a second node related to a second scale;

- a root node is a node representing a segment of the coarse segmentation;

- a leaf node is a node representing a segment of the fine segmentation;

- each node corresponds to a root node of a subtree of the hierarchic tree structure where a subset of the segments of the fine segmentation corresponds to the leaf nodes of said subtree.

- a respective index of a set of ordered indices is assigned to each of the leaf nodes, such that the indices of the leaf nodes of any subtree constitute an interval of consecutive indices.

15. The method according to any one of the claims 13 and 14, characterised in that the method further comprises the step of storing in a second storage means a mapping from each of the elements to a respective segment of the fine segmentation;

16. The method according to any one of the claims 14 and 15, characterised in that the step of generating a representation of a hierarchic tree structure comprises the step of recursively processing all segments of all segmentations starting with a segment of the coarse segmentation and until all segments are processed, where the recursive processing comprises the steps of

a) processing a first segment of a first segmentation corresponding to a first scale by attempting linking the first segment to a corresponding second segment of a second segmentation with a second scale, where the second scale is a largest of said scales which is smaller than the first scale;

b) if attempting linking the first segment succeeds continuing the processing of all segments with the second segment; otherwise

c) if there is a third segment of a third segmentation with a third scale larger than the first scale, where the third segment is already linked to the first segment, continuing the processing of all segments with the third segment; otherwise

d) if there is a fourth segment of the coarse segmentation which is not linked to any segment, continuing the processing of all segments with the fourth segment.

17. The method according to any one of the claims 14 through 16, c h a r a c t e r i s e d in that the step of generating a representation of a hierarchic tree structure comprises the step of recursively processing all segments of all segmentations by a method selected from the class of methods comprising pre-order, post-order and in-order traversals.

18. The method according to any one of the claims 14 through 17, c h a r a c t e r i s e d in that the representation of a hierarchic tree structure comprises a table where each entry corresponds to a leaf node, such that a first entry, corresponding to a first leaf node, identifies

a) a second leaf node of a smallest subtree including the first leaf node and the second leaf node, where the second leaf node has the largest index of all leaf nodes of said smallest subtree, and

b) the scale related to the root node of said smallest subtree.

19. The method according to any one of the claims 14 through 18, c h a r a c t e r i s e d in that the representation of a hierarchic tree structure comprises a table where each entry corresponds to a node, a first entry corresponding to a third node, which is a root node of a first subtree, the first entry identifies

a) a first leaf node of the first subtree with a largest index, and

b) a second leaf node of the first subtree with a smallest index.

20. The method according to any one of the claims 14 through 19, c h a r a c t e r i s e d in that the method further comprises a step of selecting at least one portion of the multidimensional representation, the step of selecting further comprising the steps of

a) selecting a first element;

b) selecting a scale;

c) identifying a first segment of the fine segmentation, where the first element is mapped to said first segment according to the stored mapping;

d) calculating a set of leaf nodes of a subtree having a root node related to the selected scale and having

a first leaf node which represents the first segment; and

e) identifying at least one element of the multidimensional representation which is mapped according to the stored mapping to a segment represented by any of the calculated leaf nodes.

21. The method according to any one of the claims 13 through 20, c h a r a c t e r i s e d in that the step of processing the multidimensional representation further comprises the steps of

a) calculating a first representation of the multidimensional representation, the first representation representing a level of variation in the multidimensional representation;

b) segmenting the first representation to obtain a watershed segmentation of the first representation, a segment of the watershed segmentation including all elements of the first representation, from which there is a path of steepest descent to a corresponding local minimum of the first representation.

22. The method according to claim 21, c h a r a c t e r i s e d in that the method further comprises a step of selecting at least one portion of the multidimensional representation, the step of selecting further comprising the steps of

a) selecting a seed segment of the watershed segmentation;

b) assigning to at least a first and second segment of the watershed segmentation a respective first and second rank, a selected one of the first and second

ranks corresponding to predetermined properties of a set of segments on a path from the seed segment to the corresponding first or second segment;

c) sorting the first and second segment according to their respective first and second ranks;

d) selecting a number of sorted segments to be included in the selected portion of the multidimensional representation;

23. The method according to any one of the claims 13 through 22, c h a r a c t e r i s e d in that the step of smoothing the respective contour surfaces of the first and second segment and the step of smoothing the contour surface of the intersection comprise the step of using a linear filter.

24. The method according to any one of the claims 13 through 23, c h a r a c t e r i s e d in that the multidimensional representation is a three-dimensional image.

25. The method according to any one of the claims 13 through 24, c h a r a c t e r i s e d in that the method further comprises the step of blurring the multidimensional representation by anisotropic diffusion to obtain at least one blurred representation corresponding to a respective scale.

26. Use of a method according to any one of the claims 1 through 25 for segmenting images obtained from a scanning according to a scanning method selected from the class of scanning methods comprising MR, CT, PET and SPECT scanning.

27. Use of a method according to any one of the claims 1 through 25 for segmenting images obtained from ultrasound scanning.

28. Use of a method according to any one of the claims 1 through 25 for segmenting multidimensional representations of geological measurements.

29. Use of a method according to any one of the claims 1 through 25 for a presentation of selectable items in a two-dimensional image.

30. A computer processor adapted to perform all steps of a method according to any one of the claims 1 through 25.

31. A computer program comprising program code means for performing all the steps of any one of the claims 1 through 25 when said program is run on a computer.

32. A computer program product comprising program code means stored on a computer readable medium for performing a method of any one of the claims 1 through 25 when said computer program product is run on a computer.

33. A computer data signal embodied in a carrier wave, comprising program code means for performing all the steps of any one of the claims 1 through 25 when said program is run on a computer.

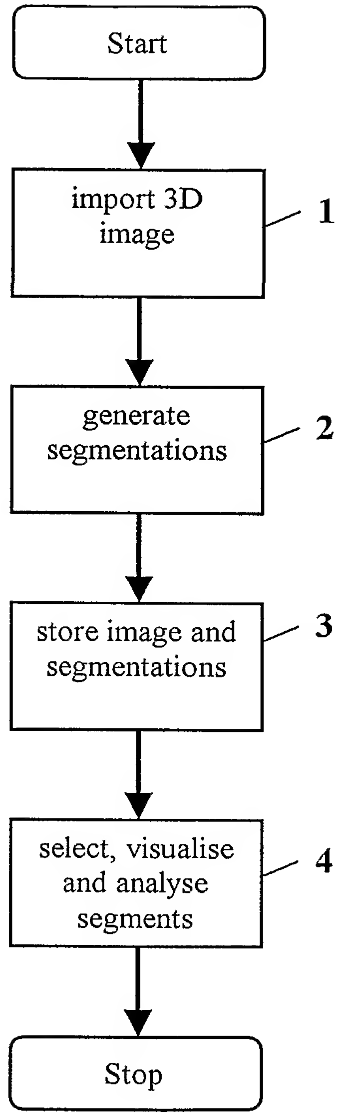

Start
import 3D image
1
generate segmentations
2
store image and segmentations
3
select, visualise and analyse segments
4
Stop

Fig. 1

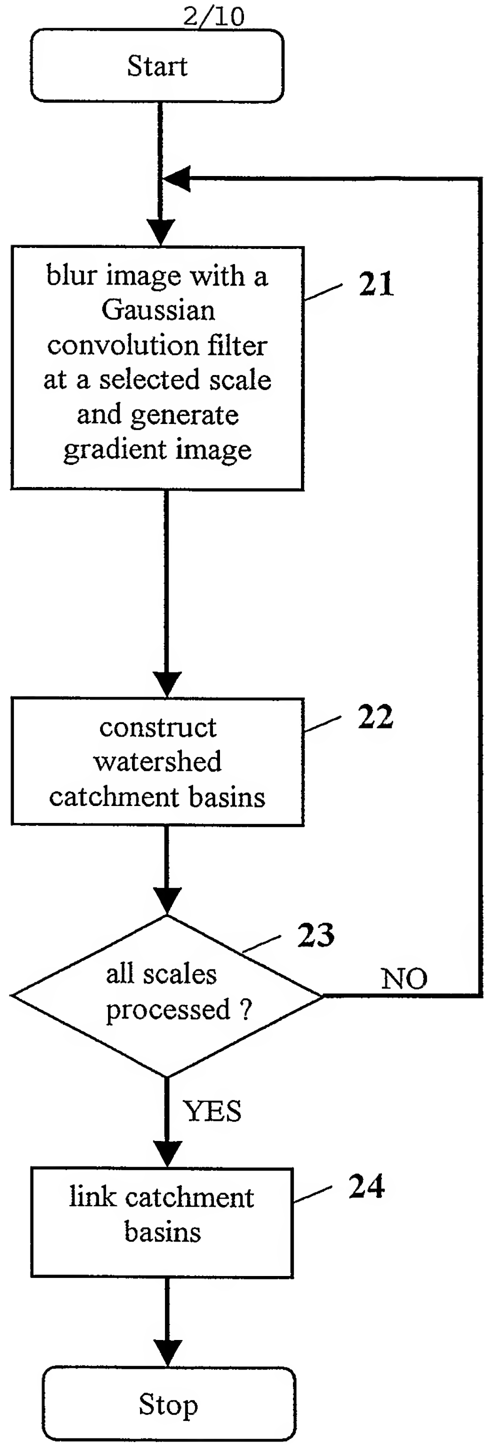

Start
blur image with a Gaussian convolution filter at a selected scale and generate gradient image
21
construct watershed catchment basins
22
all scales processed ?
23
NO
YES
link catchment basins
24
Stop

Fig. 2

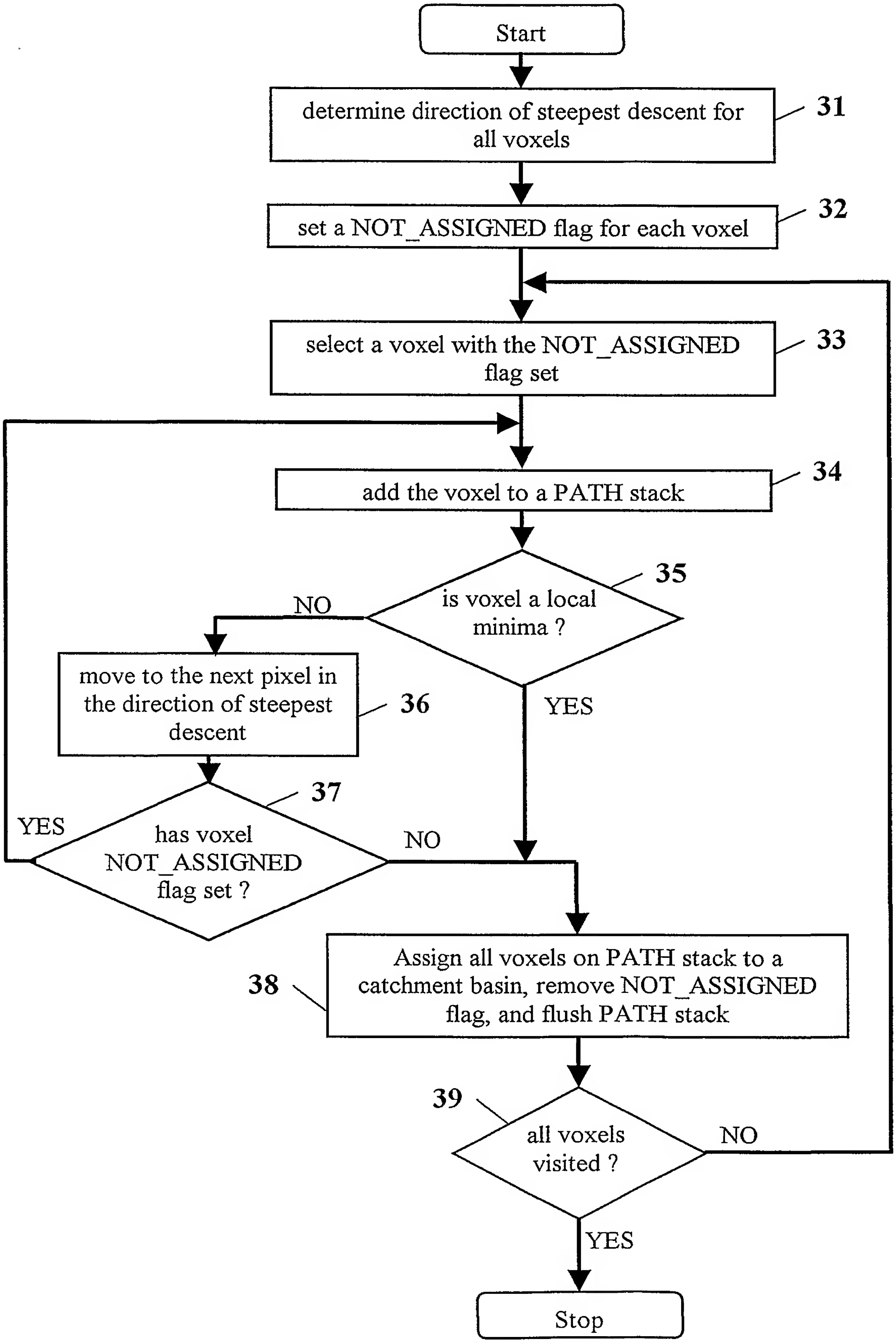

Start
determine direction of steepest descent for all voxels
31
set a NOT_ASSIGNED flag for each voxel
32
select a voxel with the NOT_ASSIGNED flag set
33
add the voxel to a PATH stack
34
is voxel a local minima ?
35
NO
YES
move to the next pixel in the direction of steepest descent
36
has voxel NOT_ASSIGNED flag set ?
37
YES
NO
Assign all voxels on PATH stack to a catchment basin, remove NOT_ASSIGNED flag, and flush PATH stack
38
all voxels visited ?
39
NO
YES
Stop

Fig. 3

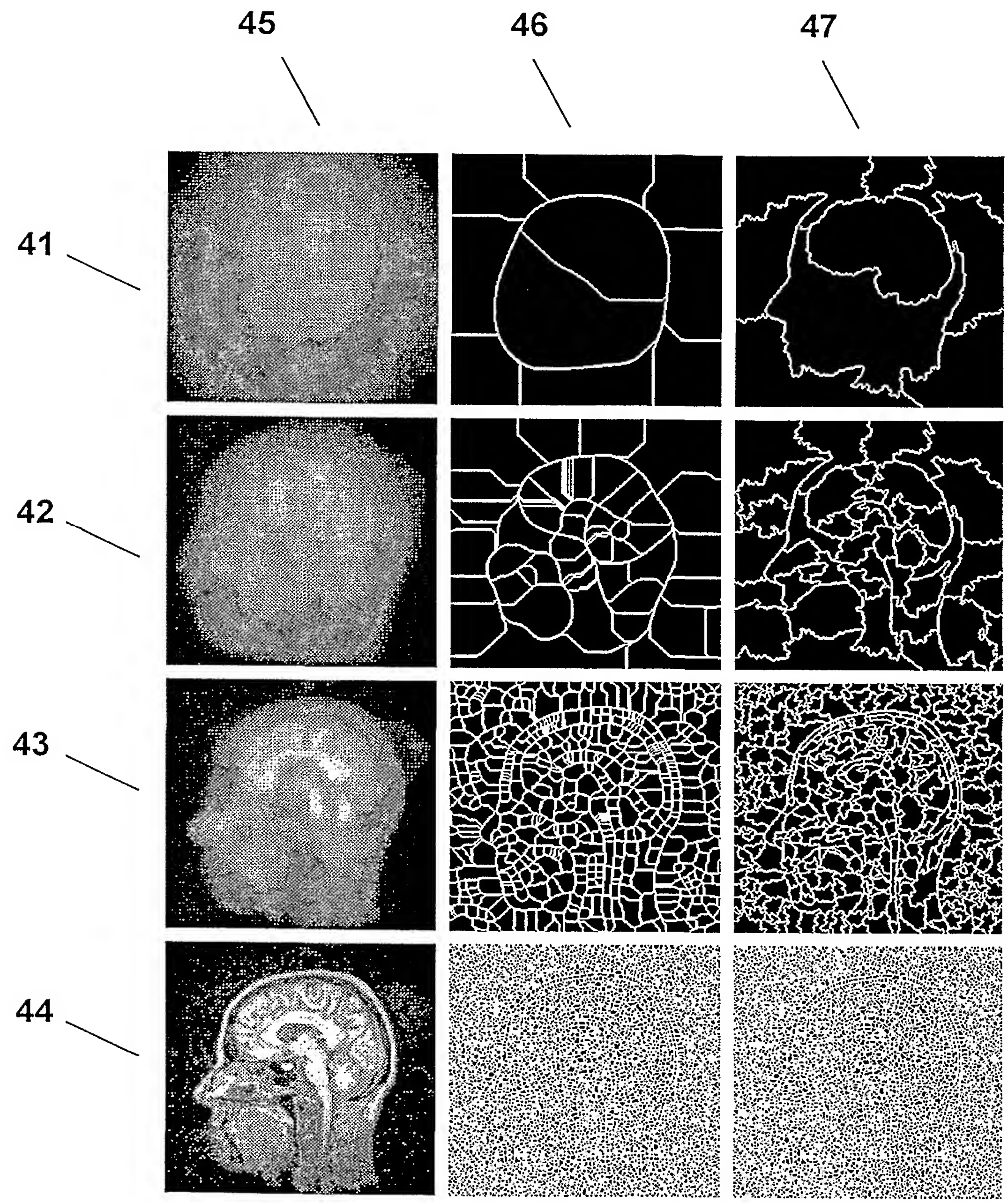


Fig. 4

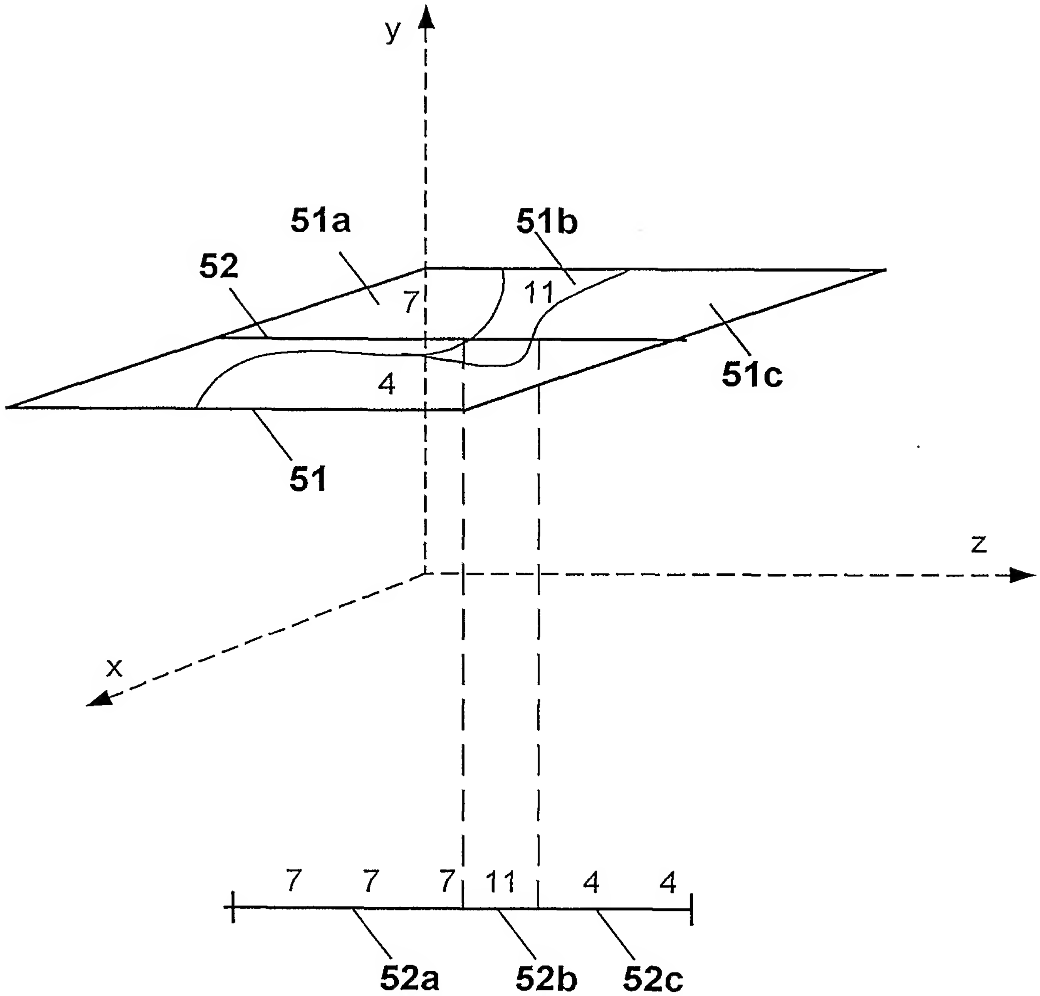


Fig. 5

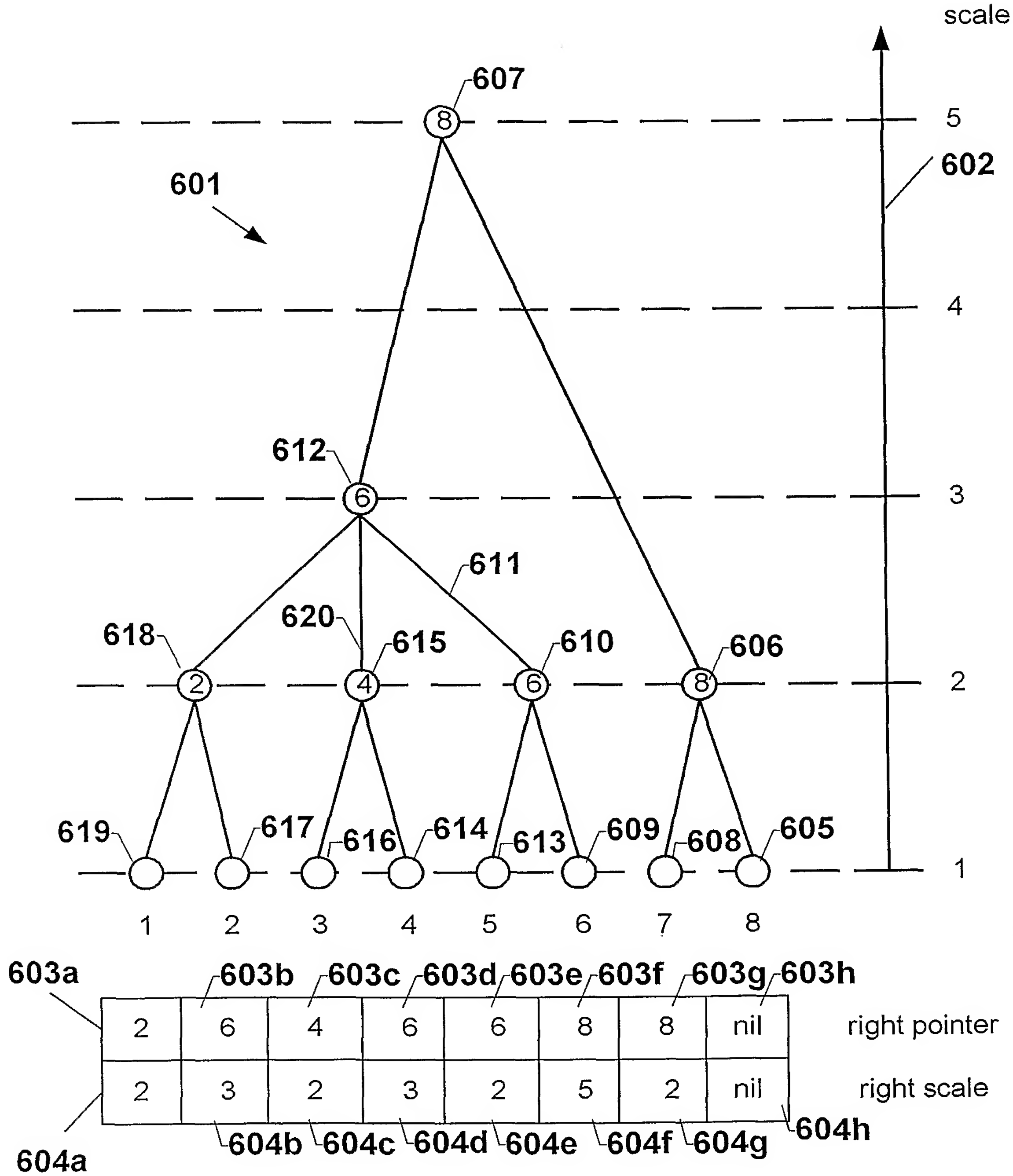

scale
607
8
5
601
602
4
612
6
3
611
618
620
615
610
606
2
4
6
8
2
619
617
616
614
613
609
608
605
1
1
2
3
4
5
6
7
8
603a
603b
603c
603d
603e
603f
603g
603h
2
6
4
6
6
8
8
nil
right pointer
2
3
2
3
2
5
2
nil
right scale
604a
604b
604c
604d
604e
604f
604g
604h

Fig. 6

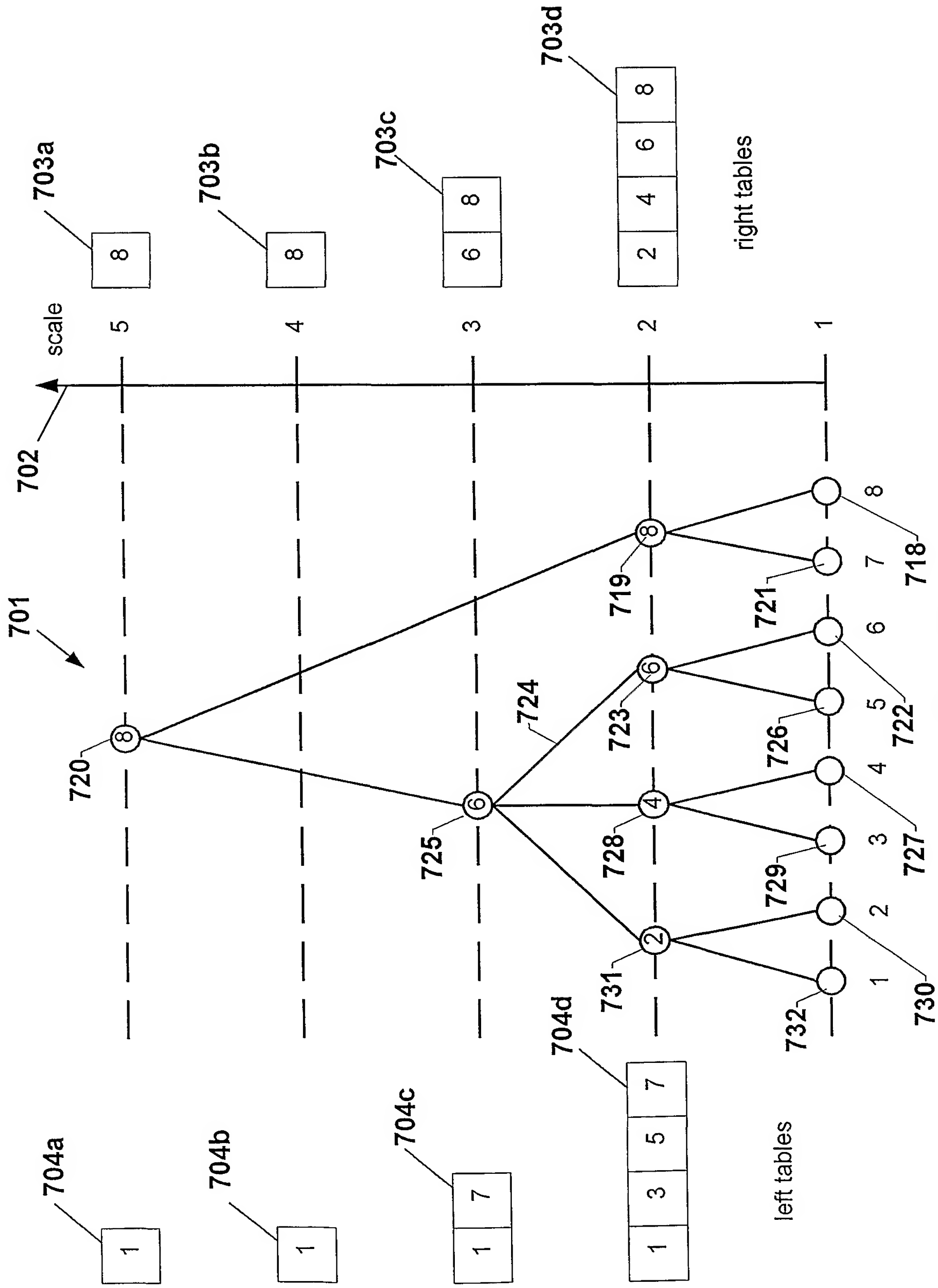

701
702
scale
5
4
3
2
1
703a
703b
703c
703d
right tables
704a
704b
704c
704d
left tables
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732

Fig. 7

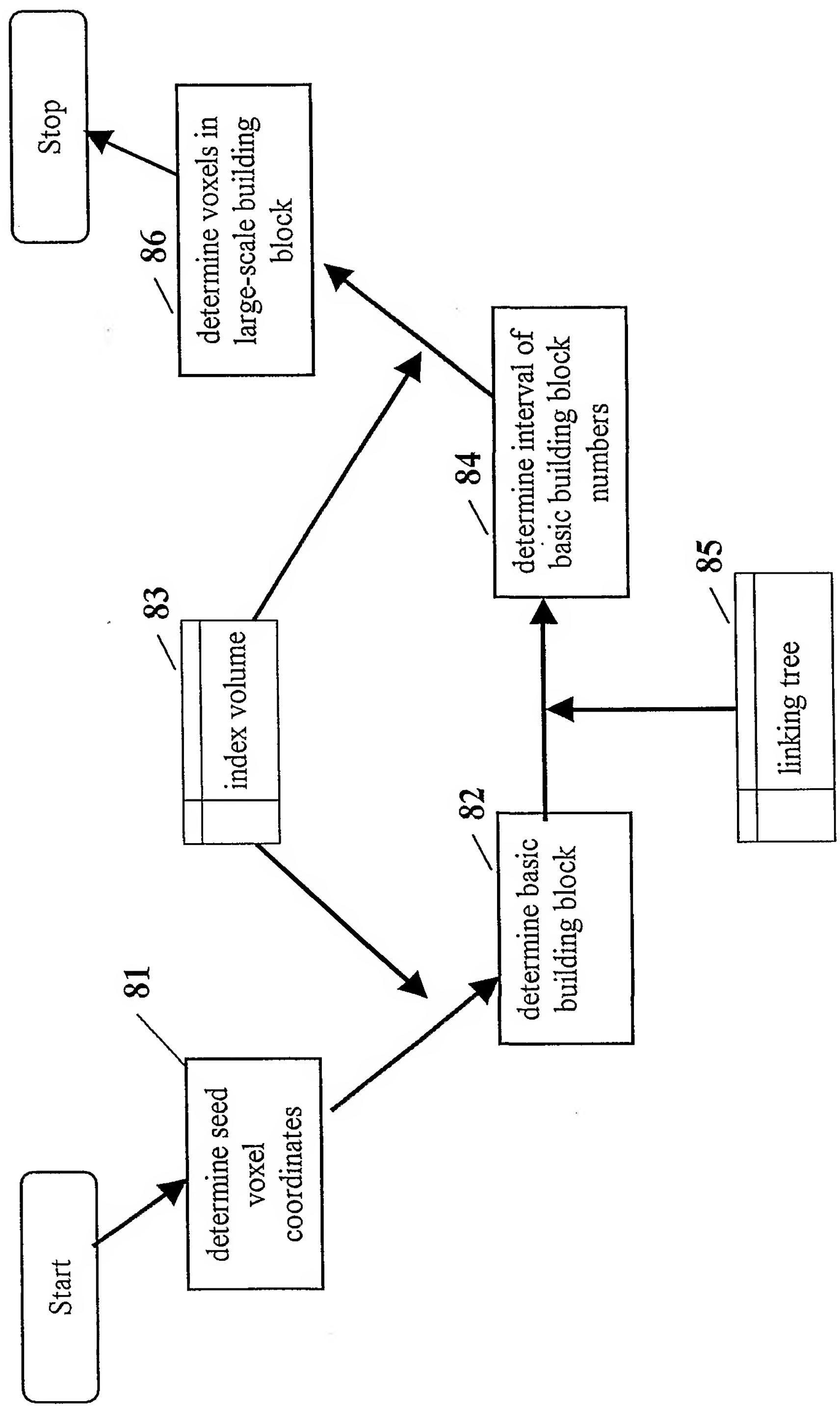

Start
determine seed voxel coordinates
81
determine basic building block
82
index volume
83
determine interval of basic building block numbers
84
linking tree
85
determine voxels in large-scale building block
86
Stop

Fig. 8

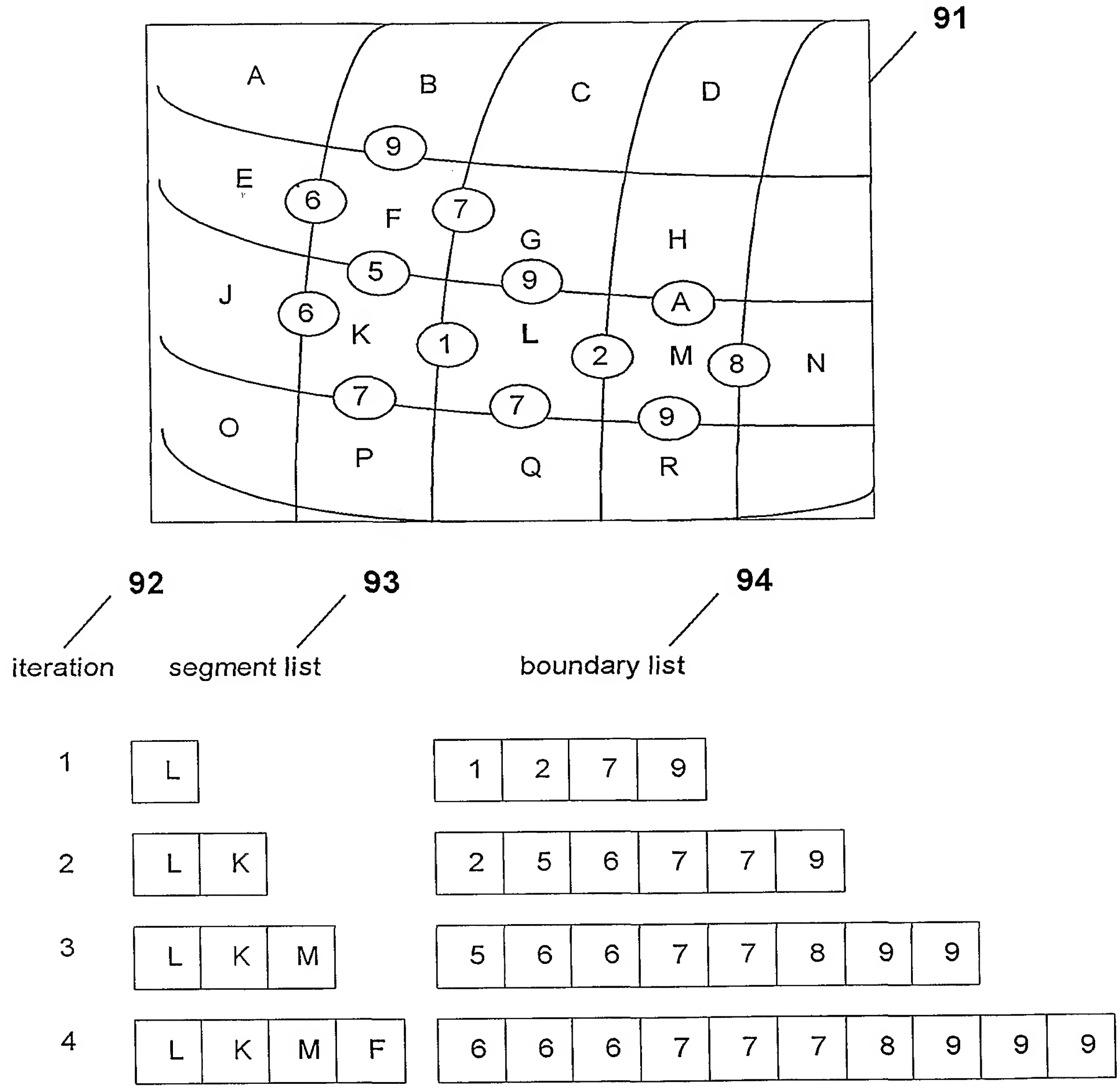

91
A
B
C
D
9
E
6
F
7
G
H
5
9
A
J
6
K
1
L
2
M
8
N
7
7
9
O
P
Q
R
92
93
94
iteration
segment list
boundary list
1
L
1 2 7 9
2
L K
2 5 6 7 7 9
3
L K M
5 6 6 7 7 8 9 9
4
L K M F
6 6 6 7 7 7 8 9 9 9

Fig. 9

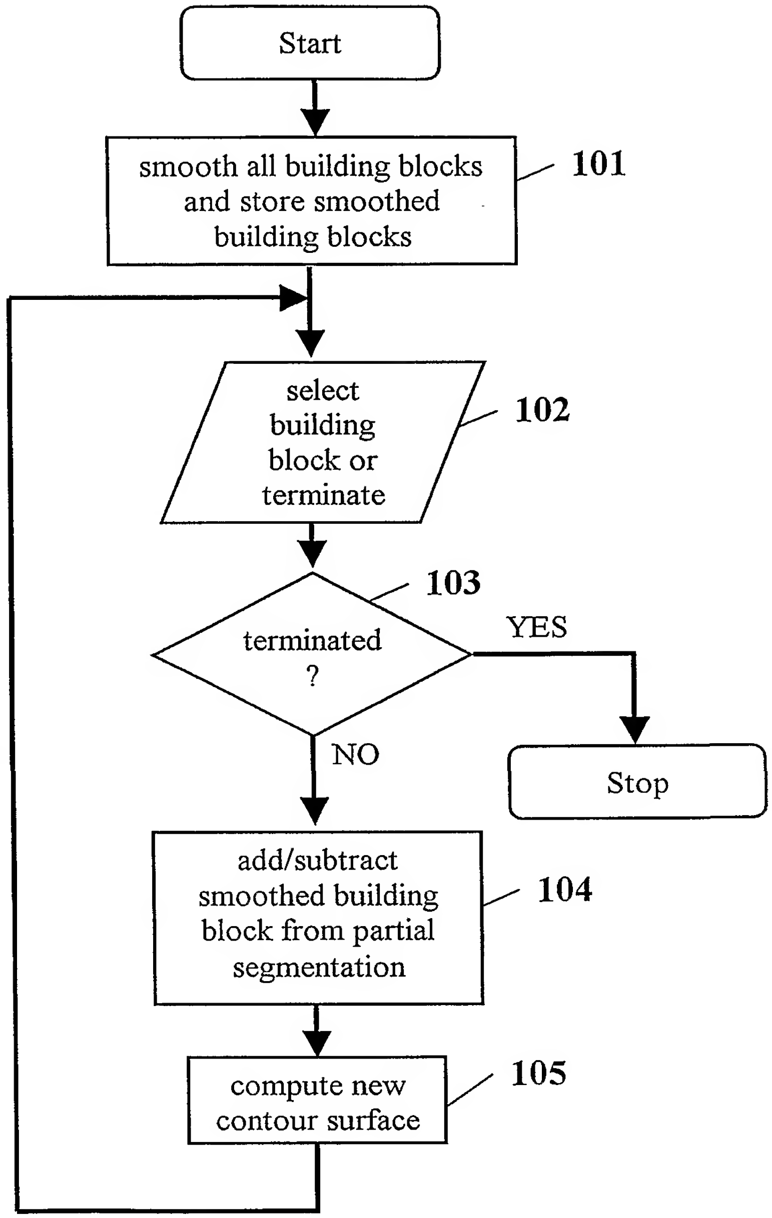

Start
smooth all building blocks and store smoothed building blocks
101
select building block or terminate
102
terminated ?
103
YES
NO
Stop
add/subtract smoothed building block from partial segmentation
104
compute new contour surface
105

Fig. 10